Technology Evaluation

# Price Gathering and Processing, API, and Automated Trading

|  |  |  |
| --- | --- | --- |
| **Technology** | **Positive** | **Negative** |
| Java | 1. Familiar 2. Suitable 3. Mature 4. Large amount of support and documentation 5. License covered 6. Platform independent | 1. Arguably inferior on Windows |
| C# | 1. Familiar 2. Suitable 3. Mature 4. Large amount of support and documentation 5. License covered 6. Arguably superior on Windows | 1. Windows only |
| Technology new to the developer | 1. Various pros | 1. Various cons 2. Unfamiliar |

## Overall decision

Java and C# have both been highlighted as technologies suitable for the task out of the technologies that I have already worked with. C++ would be worth consideration if there was an embedded system included within the project, but since there is not it seems less appropriate. Due to the project’s dependence on an unfamiliar system – the GDAX API – and the need to research into how a price can be predicted from the data available, I feel that working with a familiar technology makes sense. There will inevitably be unforeseen issues that need to be dealt with and adding the need to learn the syntax of a new programming language is illogical with the fixed time constraints of the project. Another issue with using a programming language that I am unfamiliar with would be the need to potentially pay a license fee and/or purchase an integrated development environment to develop the project in.

Given that Java and C# are very similar in terms of the positives – both are languages that I am familiar with, both have the capability of performing the task required, both are mature enough to have had a large number of issues already removed with a large amount of support and documentation of known issues and both have free licenses for my intended purpose. This leaves the decision on the following question; should the application be platform independent, or should it take advantage of C#’s arguable superiority on a Windows system? Given that approximately 88.8%[[1]](#footnote-1) of the world use Windows, it wouldn’t be illogical to focus on that group if it had a cost to target the other 11.2%, but there isn’t a cost, so development in Java is the logical decision.

# Data Storage

|  |  |  |
| --- | --- | --- |
| **Technology** | **Positive** | **Negative** |
| Oracle | 1. Familiar 2. Suitable 3. Mature 4. Large amount of support and documentation 5. License covered 6. Storage on campus | 1. Unpreventable downtime and data loss |
| Neo4J | 1. Familiar 2. Suitable 3. Mature 4. Large amount of support and documentation 5. License covered 6. Able to prevent downtime and data loss to some extent | 1. Must source own storage |
| Other technology |  |  |

Overall decision: will decide when I am closer to implementation.

1. <https://goo.gl/sBmZVR> [↑](#footnote-ref-1)